Column Generation Method

Frédéric Giroire

Column Generation in Two Words

- A framework to solve large problems.
- Main idea:
  - not considering explicitly the whole set of variables.
  - decompose the problem into a master and a subproblem (or pricing problem), use the pricing problem to generate a "good" column = variable.

$\rightarrow$ allows to solve in practice hard problems with an exponential number of variables.

Column Generation - History

- Ford and Fulkerson (1958): A first suggestion to deal implicitly with the variables of a multicommodity flow problem.
- Dantzig and Wolfe (1961): Develop a strategy to extend a linear program columnwise
- Gilmore and Gomory (1961, 1963): First implementation as part of an efficient heuristic algorithm for solving the cutting stock problem
- Desrosiers, Soumis and Desrochers (1984): Embedding of column generation within a LP based B&B framework for solving a vehicle routing problem

Column Generation (CG)

- A set of columns, $a \in \mathcal{A} \subset \mathrm{R}^m$, $c_a \in \mathrm{R}$, $b \in \mathrm{R}^m$
- Large-scale primal and dual problems:

$$
(\mathrm{P}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}} c_a\, x_a & \\ \sum_{a \in \mathcal{A}} a\, x_a & \leq b \\ x_a \geq 0 & \qquad a \in \mathcal{A} \end{aligned}
\qquad\qquad
(\mathrm{D}) \quad \begin{aligned} \min \pi\, b & \\ \pi\, a \geq c_a & \qquad a \in \mathcal{A} \\ \pi \geq 0 & \end{aligned}
$$

- $\mathcal{A}$ too large: impossible (or impractical) to solve at once
- Column Generation: select $\mathcal{A}' \subseteq \mathcal{A}$, solve Restricted Master Problems (RMP)

$$
(\mathrm{P}_{\mathcal{A}'}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}'} c_a\, x_a & \\ \sum_{a \in \mathcal{A}'} a\, x_a & \leq b \\ x_a \geq 0 & \qquad a \in \mathcal{A}' \end{aligned}
\qquad\qquad
(\mathrm{D}_{\mathcal{A}'}) \quad \begin{aligned} \min \pi\, b & \\ \pi\, a \geq c_a & \qquad a \in \mathcal{A}' \\ \pi \geq 0 & \end{aligned}
$$

Column Generation (CG)

- A set of columns, $a \in \mathcal{A} \subset \mathrm{R}^m$, $c_a \in \mathrm{R}$, $b \in \mathrm{R}^m$
- Large-scale primal and dual problems:

$$(\mathrm{P}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}} c_a\, x_a & \\ \sum_{a \in \mathcal{A}} a\, x_a & \leq b \\ x_a \geq 0 \qquad & a \in \mathcal{A} \end{aligned} \qquad\qquad (\mathrm{D}) \quad \begin{aligned} & \min \pi\, b \\ & \pi\, a \geq c_a \qquad a \in \mathcal{A} \\ & \pi \geq 0 \end{aligned}$$

- $\mathcal{A}$ too large: impossible (or impractical) to solve at once
- Column Generation: select $\mathcal{A}' \subseteq \mathcal{A}$, solve Restricted Master Problems (RMP)

$$(\mathrm{P}_{\mathcal{A}'}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}'} c_a\, x_a & \\ \sum_{a \in \mathcal{A}'} a\, x_a & \leq b \\ x_a \geq 0 \qquad & a \in \mathcal{A}' \end{aligned} \qquad\qquad (\mathrm{D}_{\mathcal{A}'}) \quad \begin{aligned} & \min \pi\, b \\ & \pi\, a \geq c_a \qquad a \in \mathcal{A}' \\ & \pi \geq 0 \end{aligned}$$

Column Generation (CG)

- A set of columns, $a \in \mathcal{A} \subset \mathrm{R}^m$, $c_a \in \mathrm{R}$, $b \in \mathrm{R}^m$
- Large-scale primal and dual problems:

$$(\mathrm{P}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}} c_a \, x_a & \\ \sum_{a \in \mathcal{A}} a \, x_a & \leq b \\ x_a \geq 0 \qquad & a \in \mathcal{A} \end{aligned} \qquad\qquad (\mathrm{D}) \quad \begin{aligned} \min \pi \, b & \\ \pi \, a \geq c_a \qquad & a \in \mathcal{A} \\ \pi \geq 0 & \end{aligned}$$

- $\mathcal{A}$ too large: impossible (or impractical) to solve at once
- Column Generation: select $\mathcal{A}' \subseteq \mathcal{A}$, solve Restricted Master Problems (RMP)

$$(\mathrm{P}_{\mathcal{A}'}) \quad \begin{aligned} \max \sum_{a \in \mathcal{A}'} c_a \, x_a & \\ \sum_{a \in \mathcal{A}'} a \, x_a & \leq b \\ x_a \geq 0 \qquad & a \in \mathcal{A}' \end{aligned} \qquad\qquad (\mathrm{D}_{\mathcal{A}'}) \quad \begin{aligned} \min \pi \, b & \\ \pi \, a \leq c_a \qquad & a \in \mathcal{A}' \\ \pi \geq 0 & \end{aligned}$$

- primal feasible $x^\star$ and dual infeasible $\pi^\star$

Column Generation (2)

- Then solve pricing (or separation) problem

$$(\mathsf{P}_{\pi^\star}) \qquad\qquad \max\{c_a - \pi^\star a : a \in \mathcal{A}\}$$

for some $a \in \mathcal{A} \setminus \mathcal{A}'$ or optimality certificate $\pi^\star a \geq c_a \quad \forall a \in \mathcal{A}$

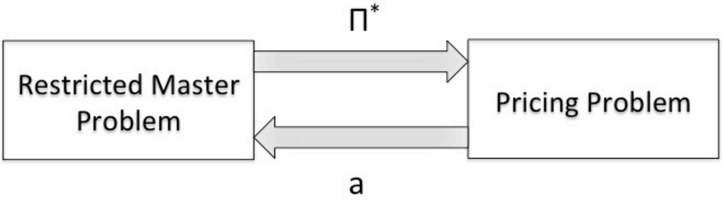
Column Generation (2)

- Then solve pricing (or separation) problem

$$(\mathsf{P}_{\pi^\star}) \qquad\qquad \max\{c_a - \pi^\star a : a \in \mathcal{A}\}$$

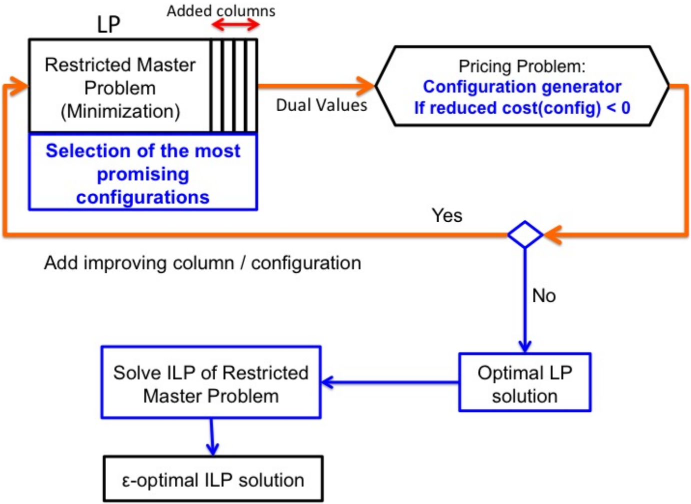
for some $a \in \mathcal{A} \setminus \mathcal{A}'$ or optimality certificate $\pi^\star a \geq c_a \quad \forall a \in \mathcal{A}$

- Very simple idea, very simple implementation (in principle)

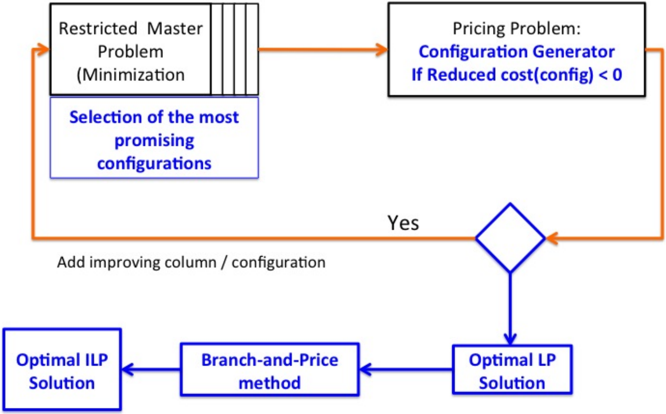


... yet surprisingly effective in many applications
... provided that (P) can be efficiently solved

Column Generation Flowchart #1 (Minimization)



Column Generation Flowchart #2 (Minimization)

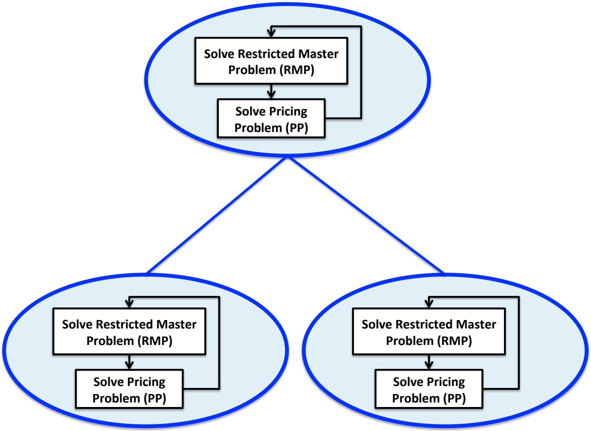


How to Get an Integer Solution (Minimization)

- What we are looking for: $z^\star_{\mathrm{ILP}}$
- CG $\rightsquigarrow z^\star_{\mathrm{LP}}$, a lower bound on $z^\star_{\mathrm{ILP}}$
- Solve the last RMP as an ILP $\rightsquigarrow \tilde{z}_{\mathrm{ILP}} \neq z^\star_{\mathrm{ILP}}$

Lower bound		Upper bound
$z^\star_{\mathrm{LP}}$	$z^\star_{\mathrm{ILP}}$	$\tilde{z}_{\mathrm{ILP}}$
Optimal LP solution	Optimal ILP solution	Heuristic ILP solution

Branch-and-Price Methods



Branch-and-Price Methods (2)

- On the variables of the Master Problem
  - Select x_a, and create two branches: $x_a = 0$ and $x_a = 1$
  - Continue with the other variables...
  - Can be improved by using cuts
- Different branching schemes...

Example 1:
Cutting-Stock Problem

The cutting stock problem

- A paper company with a supply of large rolls of paper, of width W
- Customer demand is for smaller widths of paper
  $\rightsquigarrow b_i$ rolls of width $w_i \leq W$, $i = 1, 2, \ldots\ldots m$ need to be produced.
- Smaller rolls are obtained by slicing large rolls
- Example: a large roll of width 70 can be cut into 3 rolls of width $w_1 = 17$ and 1 roll of width $w_2 = 15$, with a waste of 4.

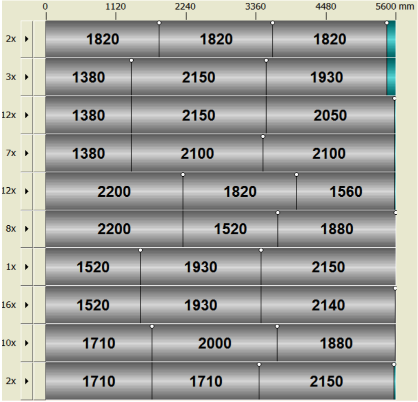
minimize the waste !

An Example

The width of large rolls: 5600mm. The width and demand of customers:

Width	1380	1520	1560	1710	1820	1880	1930	2000	2050	2100	2140	2150	2200
Demand	22	25	12	14	18	18	20	10	12	14	16	18	20

An optimal solution:



Kantorovich Model

Variables:

$y_k = 1$ if roll k is used, 0 otherwise

$x_{ik} = \#$ of times item i of width w_i, is cut in roll k

[ex: write LP]

Drawbacks of the Kantorovich Model

- However, the IP formulation (P_1) is inefficient both from computational and theoretical point views.
- The main reason is that the linear program (LP) relaxation of (P_1) is poor. Actually, the LP bound of (P_1) is

$$z^{\star}_{\mathsf{LP}} = \sum_{k \in \mathcal{K}} y_k = \sum_{k \in \mathcal{K}} \sum_{i=1}^{m} \frac{w_i \, x_i^k}{W} = \sum_{i=1}^{m} w_i \sum_{k \in \mathcal{K}} \frac{x_i^k}{W} = \sum_{i=1}^{m} \frac{w_i \, b_i}{W} \qquad (1)$$

- Question: Is there an alternative?

Formulation of Gilmore and Gomory

- Let
  λ_p = number of times pattern p is used
  a_i^p = number of times item i is cut in pattern p
- For example, the fixed width of large rolls is $W = 100$ and the demands are $b_i = 100, 200, 300, w_i = 25, 35, 45$ $(i = 1, 2, 3)$.
- The large roll can be cut into

  Pattern 1: 4 rolls each of width $w_1 = 25 \rightsquigarrow a_1^1 = 4$
  Pattern 2: 1 roll with width $w_1 = 25$ and 2 rolls each of width $w_2 = 35 \rightsquigarrow a_1^2 = 1, a_2^2 = 2$
  Pattern 3: 2 rolls with width $w_3 = 45 \rightsquigarrow a_3^3 = 2.$

Gilmore - Gomory Model & Master Problem

Cutting pattern p:
described by the vector $(a_1^p, a_2^p, ..., a_m^p)$, where a_i^p represents the number of rolls of width w_i obtained in cutting pattern p.

Variables:
$\lambda_p =$ # rolls to be cut according to cutting pattern p .

[ex: write LP]

Gilmore - Gomory Model & Master Problem

Cutting pattern p:
described by the vector $(a_1^p, a_2^p, ..., a_m^p)$, where a_i^p represents the number of rolls of width w_i obtained in cutting pattern p.

Variables:
$\lambda_p =$ # rolls to be cut according to cutting pattern p .

[ex: write LP]

- Each column represents a cutting pattern.
- How many columns (cutting patterns) are there? It could be as many as $\frac{m!}{\bar{k}!(m-\bar{k})!}$ where $\bar{k}$ is the average number of items in each cutting patterns. **Exponentially large!.**

Cutting Pattern & Pricing Problem

- pth pattern, can be represented by a column vector a^p whose ith entry indicates how many rolls of width w_i are produced by that pattern.
- Pattern discussed earlier can be represented by the vector:

$$(3, 1, 0, ..., 0)$$

- For a vector $(a_1^p; a_2^p, ..., a_m^p)$ to be a representation of a feasible pattern, its components must be nonnegative integers, and we must also have:

$$\sum_{i=1}^{m} w_i \, a_i^p \leq W.$$

Initial Solution

First step, set up column generation iteration starting point. Actually, the initial basis matrix can be constructed in a trivial way. For example, let W = 10 and the customer demand: small roll widths $w_1 = 3$ and $w_2 = 2$.

The following two choices are both valid:

- Choice 1:

$$a^{p1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad a^{p2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Choice 2:

$$a^{p1} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}; \quad a^{p2} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

Definition of the Restricted Master Problem

$P' \subseteq P$ a small subset of cutting patterns

$$\begin{aligned}
\min & \sum_{p \in P'} \lambda_p & \\
\text{s.t.} & \sum_{p \in P'} a_i^p \lambda_p \geq 0 & i = 1, 2, \ldots m \\
& \lambda_p \geq 0 & p \in P'
\end{aligned}$$

Note:

- m types of small roll customers requested
- $|P'|$ number of patterns generated so far
- column generation $\rightsquigarrow$ tool for solving the linear programming relaxation.

How to Iterate?

Suppose that we have a basis matrix B for the restricted master problem and an associated basic feasible solution, and that we wish to carry out the next iteration for the column generation.

Because the coefficient of every variable λ_p is 1, every component of the vector c_B is equal to 1.

Next, instead of computing the reduced cost $\bar{c}_B = 1 - c_B B^{-1} a^p$ associated with every column a^p, we consider the problem of minimizing $(\bar{c}_B = 1 - c_B B^{-1} a^p)$ over all p.
This is the same as maximizing $c_B B^{-1} a^p$ over all p.

- If the maximum is ≤ 1, all reduced costs are nonnegative and we have an optimal solution.
- If the maximum is > 1, column a^p corresponding to a maximizing p has negative reduced cost and enters the basis.

Pricing Problem

Recall that $u = c_B\, B^{-1}$

$$\max \sum_{i=1}^{m} u_i a_i$$

s.t. [ex: write constraint]

$$a_i \geq 0 \qquad\qquad i = 1, 2, \ldots.., m$$

$$a_i \in \mathbb{Z}^+ \qquad\qquad i = 1, 2, \ldots.., m.$$

An Example

Suppose for the paper company, a big roll of paper is $W = 218cm$.
The customers of the company want:

- 44 small rolls of length 81cm
- 3 small rolls of length 70cm
- 48 rolls of length 68 cm

That is,

$$w = \begin{pmatrix} 81 \\ 70 \\ 68 \end{pmatrix}, \quad b = \begin{pmatrix} 44 \\ 3 \\ 48 \end{pmatrix}$$

Initialization

Step 1, to generate the initial basis matrix:

$$a^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad a^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad a^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

First restricted master problem:

$$\begin{array}{lll} \min & \lambda_1 + \lambda_2 + \lambda_3 & \\ \text{s.t.} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \geq \begin{bmatrix} 44 \\ 3 \\ 48 \end{bmatrix} & \\ & \lambda_p \geq 0 & p = 1, 2, \ldots.., 5. \end{array}$$

Solving the Restricted Master and Pricing Problems

Solution of the restricted master problem

$\rightsquigarrow \quad u = c_B\, B^{-1} = (1, 1, 1).$

Therefore, the first pricing problem is written

$$
\begin{array}{lccccr}
\max & a_1 & +a_2 & +a_3 & & \\
\text{subject to:} & 81a_1 & +70a_2 & +68a_3 & \leq 218 & \\
 & a_i & \geq 0 & & & i = 1, 2, 3 \\
 & a_i & \in \mathbb{Z}^+ & & & i = 1, 2, 3
\end{array}
$$

The optimal solution is $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}$ with optimal value $3 > 1$.

The Second Restricted Master Problem

$$\min \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4$$

$$\text{s.t.} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} \geq \begin{bmatrix} 44 \\ 3 \\ 48 \end{bmatrix}$$

$$\lambda_p \geq 0 \qquad\qquad p = 1, 2, ..., 5.$$

After solving, we get $u = (1, 1, 0.33)$.

Solving the Second Pricing Problem

By solving the restricted master problem, we can get $u = (1, 1, 0.33)$. Therefore, the second pricing problem is

$$
\begin{array}{llllll}
\max & a_1 & +a_2 & +0.33a_3 & & \\
\text{s.t.} & 81a_1 & +70a_2 & +68a_3 & \leq 218 & \\
 & a_i & \geq 0 & & & i = 1, 2, 3 \\
 & a_i & \in \mathbb{Z}^+ & & & i = 1, 2, 3
\end{array}
$$

The optimal solution is

$$
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}
$$

with optimal value $3 > 1$.

The Third Restricted Master Problem

$$\min \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} \geq \begin{bmatrix} 44 \\ 3 \\ 48 \end{bmatrix}$$

$$\lambda_p \geq 0 \qquad p = 1, 2, \ldots.., 5.$$

After solving the above program, we get $u = (1, 0.33, 0.33)$.

Solving the Third Pricing Problem

By solving the restricted master problem, we can get $u = (1, 0.33, 0.33)$. Therefore, the second pricing problem is

$$
\begin{array}{llllll}
\max & a_1 & +0.33a_2 & +0.33a_3 & & \\
\text{s.t.} & 81a_1 & +70a_2 & +68a_3 & \leq 218 & \\
 & a_i & \geq 0 & & & i = 1, 2, 3 \\
 & a_i & \in \mathbb{Z}^+ & & & i = 1, 2, 3
\end{array}
$$

The optimal solution is $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$ with optimal value $2 > 1$.

Restricted Master Problem # 4

$$\min \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6$$

subject to:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix} \geq \begin{bmatrix} 44 \\ 3 \\ 48 \end{bmatrix}$$

$$\lambda_p \geq 0 \qquad p = 1, 2, \ldots.., 6$$

After solving the above program, we get $u = (0.5, 0.33, 0.33)$.

Pricing Problem # 4

By solving the restricted master problem, we can get $u = (0.5, 0.33, 0.33)$. Therefore, the second pricing problem is

$$
\begin{array}{llllll}
\max & 0.5\, a_1 & +0.33\, a_2 & +0.33\, a_3 & & \\
\text{s.t.} & 81\, a_1 & +70\, a_2 & +68\, a_3 & \leq 218 & \\
 & a_i & \geq 0 & & & i = 1, 2, 3 \\
 & a_i & \in \mathbb{Z}^+ & & & i = 1, 2, 3
\end{array}
$$

The optimal solution is

$$
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}
$$

with optimal value $1.16 > 1$.

Restricted Master Problem # 5

$$\min \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \end{bmatrix} \geq \begin{bmatrix} 44 \\ 3 \\ 48 \end{bmatrix}$$

$$\lambda_p \geq 0 \qquad p = 1, 2, \ldots.., 7.$$

After solving the above program, we get $u = (5, 0.33, 0.25)$.

Pricing Problem # 5

Restricted master problem # 5 $\rightsquigarrow u = (0.5, 0.33, 0.25)$.
Therefore, pricing problem # 5 is

$$
\begin{array}{llllll}
\max & 0.5\, a_1 & +0.33\, a_2 & +0.25\, a_3 & & \\
\text{s.t.} & 81\, a_1 & +70\, a_2 & +68\, a_3 & \leq 218 & \\
 & a_i & \geq 0 & & & i = 1, 2, 3 \\
 & a_i & \in \mathbb{Z}^+ & & & i = 1, 2, 3
\end{array}
$$

The optimal solution is

$$
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}
$$

with optimal value 1 $\rightsquigarrow$ optimality condition is satisfied.

Optimal Solution

The optimal solution of the cutting stock problem is:

$$1\begin{bmatrix}0\\3\\0\end{bmatrix} \qquad + 10\begin{bmatrix}2\\0\\0\end{bmatrix} \qquad + 24\begin{bmatrix}1\\0\\2\end{bmatrix}$$

with optimal value 35.

Question: if the initial matrix basis is :

$$a^1 = \begin{bmatrix}0\\3\\0\end{bmatrix}; \qquad a^2 = \begin{bmatrix}2\\0\\0\end{bmatrix}; \qquad a^3 = \begin{bmatrix}0\\0\\3\end{bmatrix}$$

How many iterations you need to take before reaching an LP optimal solution?

Column Generation - Take aways

When to use Column Generation?

- Compact formulation of a ILP cannot solved efficiently, e.g. due to
  - weak LP relaxation. Can be tightened by column generation formulation.
  - Symmetric structure $\hookrightarrow$ poor performance of B&B. Column generation formulation allows us to get ride of symmetries.
- Existence of a "natural decomposition".

Column Generation - Why it works

- Master Problem $\hookrightarrow$ Very often, a general ILP problem
- Pricing Problem $\hookrightarrow$ An ILP problem with a special structure
  - Knapsack problem (e.g., the pricing problem of the cutting stock problem).
  - Shortest path problem.
  - Maximum flow problem.
  - Any combinatorial problem.

The success of applying column generation often relies on efficient algorithms to solve the pricing problem.

- Moreover, column generation $\hookrightarrow$ decomposition: Natural interpretation often allows us to take care easily of additional constraints.